

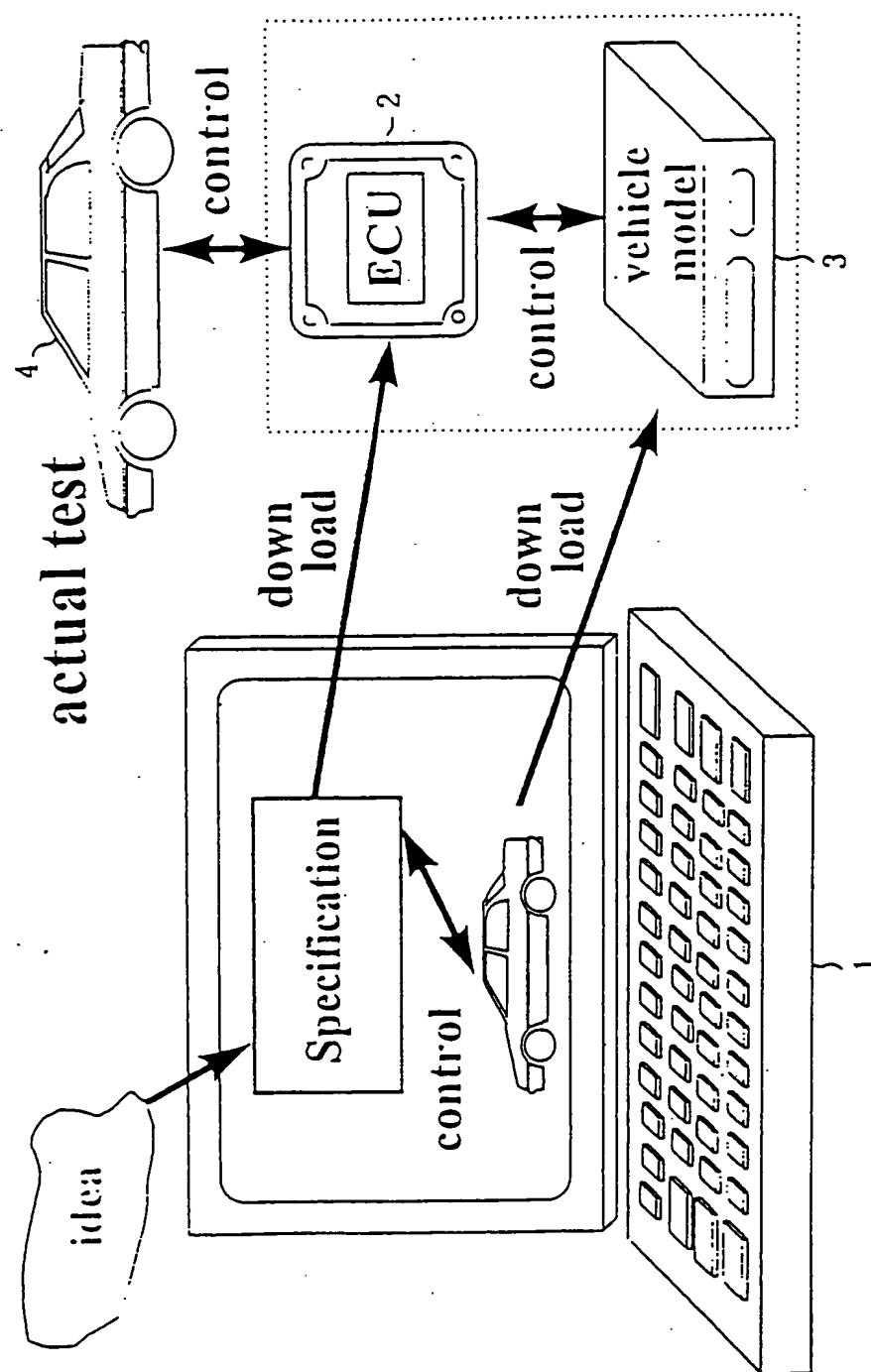
Fig. 1

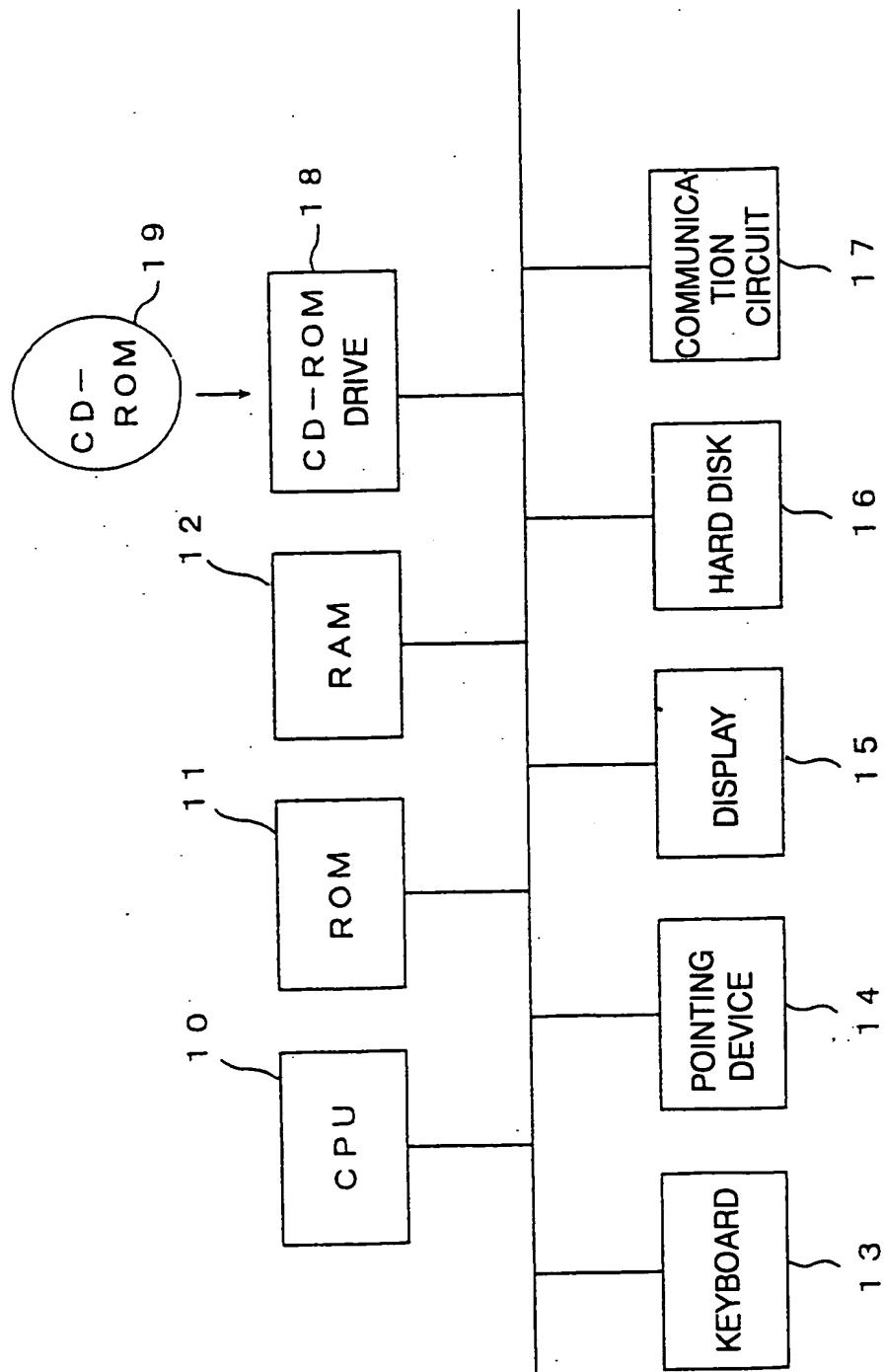
Fig. 2

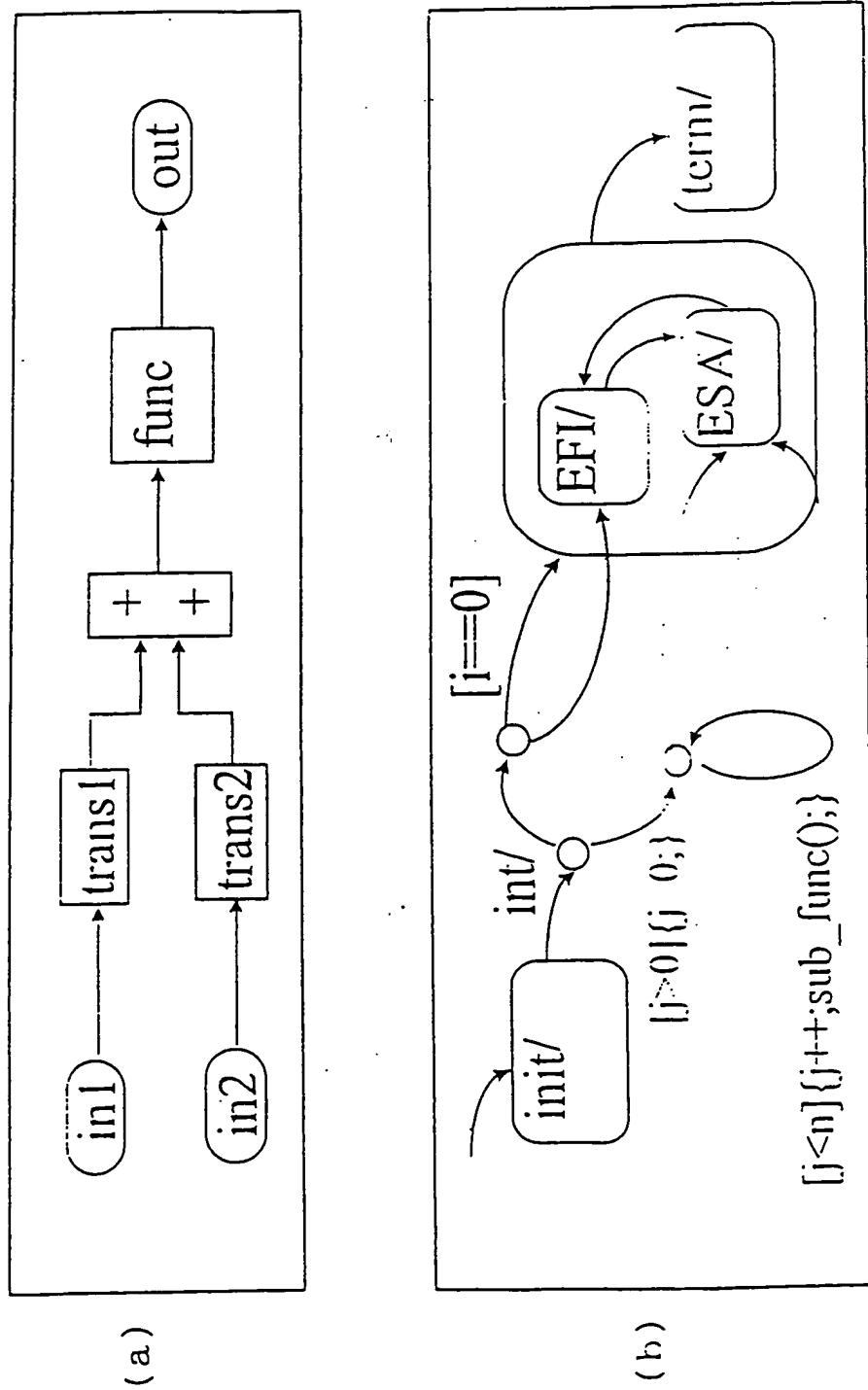
Fig. 3

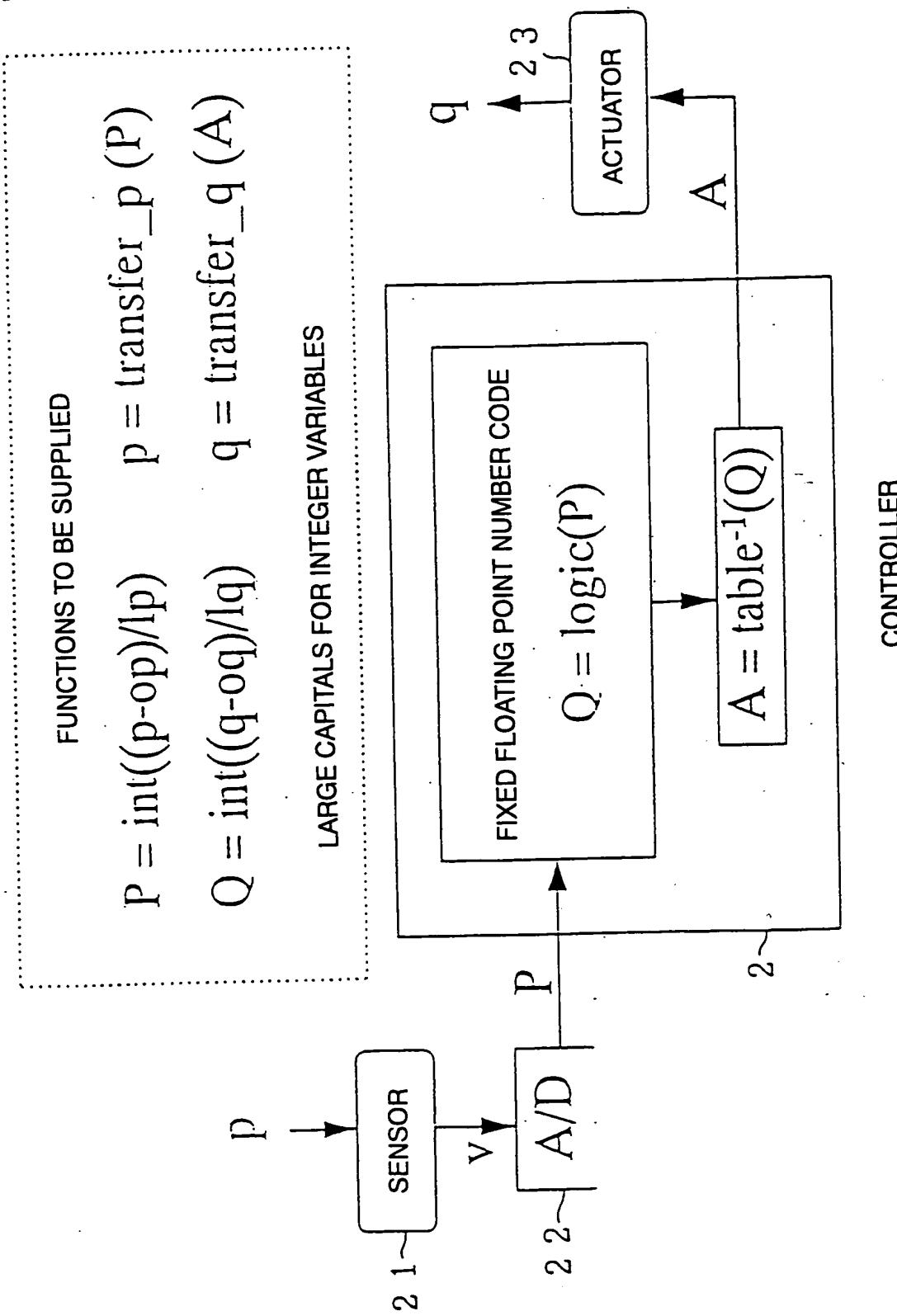
Fig. 4

Fig. 5

$$\begin{aligned}y(k) = & 0.7888 y(k-1) + \\& 0.1784 y(k-2) - \\& 0.1000 y(k-3) - \\& 0.0010 u(k) + \\& 0.0150 u(k-1) - \\& 0.0040 u(k-2) - \\& 0.0020 u(k-3)\end{aligned}$$

EXAMPLE OF AN EXPRESSION INCLUDING
A FLOATING POINT NUMBER

09/673504

0005 T2041 C147985 0003

Fig. 6

CONVERSION EXPRESSION
FLOATING POINT NUMBER → INTEGER
 $x_int = (x_float - \text{OFFSET})/\text{SLOPE}$

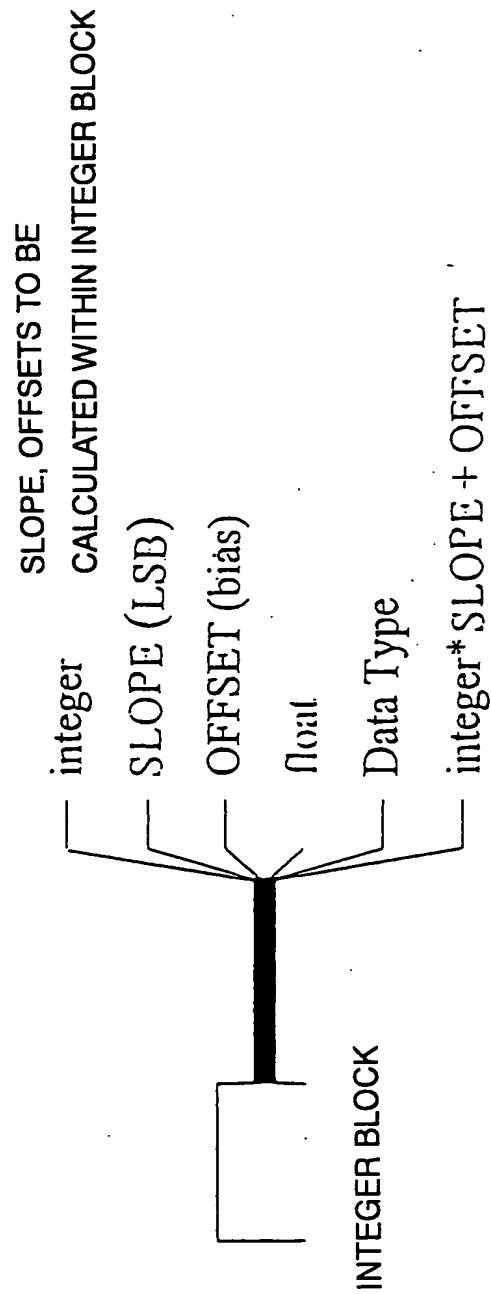
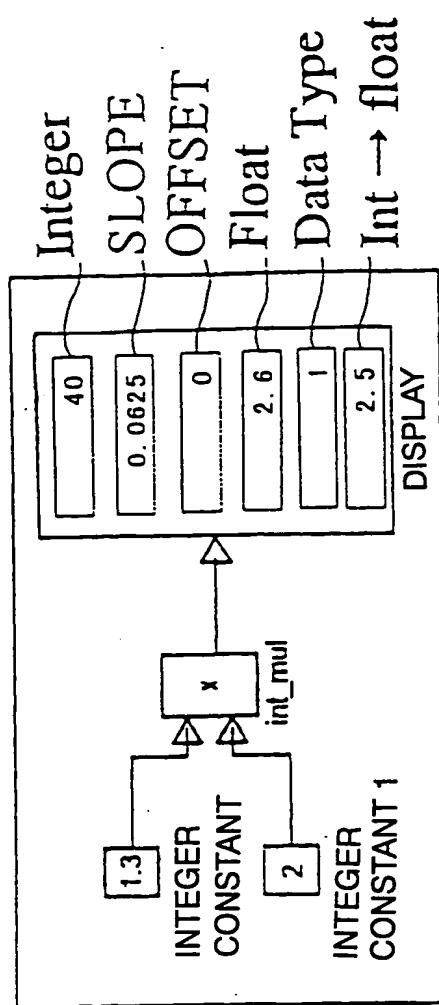


Fig. 7



```

INPUT 1 : SLOPE 0.25
          OFFSET 0
INPUT 2 : SLOPE 0.25
          OFFSET 0
    
```

Fig. 8

(1)

MULTIPLICATION

$$\frac{\text{INPUT 1}}{\text{slope1}} \times \frac{\text{INPUT 2}}{\text{slope2}}$$

$$= \frac{\text{OUTPUT}}{\frac{\text{xy} - \text{offset1} * \text{y} - \text{x} * \text{offset2} + \text{offset1} * \text{offset2}}{\text{slope1} * \text{slope2}}}$$

OUTPUT SLOPE : slope1 * slope2

OUTPUT OFFSET : offset1 * y + x * offset2 - offset1 * offset2

Fig. 9

(2)

MULTIPLICATION

INPUT 1

$$\left(\frac{(x - \text{offset}1)}{\text{slope}1} + \frac{\text{offset}1}{\text{slope}1} \right) \times \left(\frac{\left(\frac{(y - \text{offset}2)}{\text{slope}2} + \frac{\text{offset}2}{\text{slope}2} \right)}{Y_{\text{int}}} \right) = \frac{XY}{\text{slope}1 * \text{slope}2}$$

INPUT 2

DESIGNATED SLOPE : slope_0 → OUTPUT
 DESIGNATED OFFSET : offset_0 → OUTPUT
 $OFF1_{\text{int}}$: offset_1/slope_1, $OFF2_{\text{int}}$: offset_2/slope_2

MULTIPLIED INTEGER VALUE :

$$((X_{\text{int}} + OFF1_{\text{int}}) * (Y_{\text{int}} + OFF2_{\text{int}})) * \text{slope}1 * \text{slope}2 - \text{offset}_0 / \text{slope}_0$$

Fig. 10

(1)

(1) ALL INTEGER LINES ARE VECTORS CONSTITUTING OF THE BELOW

- y(1) = integer value
- y(5) = signed or unsigned
- y(2) = LSB
- y(6) = CARRY
- y(3) = OFFSET
- y(7) = ZERO
- y(4) = floating point value
- y(8) = NEGATIVE

(2) INTEGER LOGIC SUPPORT TOOL HAS BLOCKS BELOW

(1) CONVERSION TO INTEGER	(9) 2D TABLE LOOK UP
(2) CONVERSION TO FLOATING	(10) SPLIT
(3) ADDITION	(11) UNIT DELAY
(4) MULTIPLICATION	(12) INTEGER SCOPE
(5) DIVISION	
(6) SURPLUS	
(7) SHIFT	
(8) 1D TABLE LOOK UP	

Fig. 11

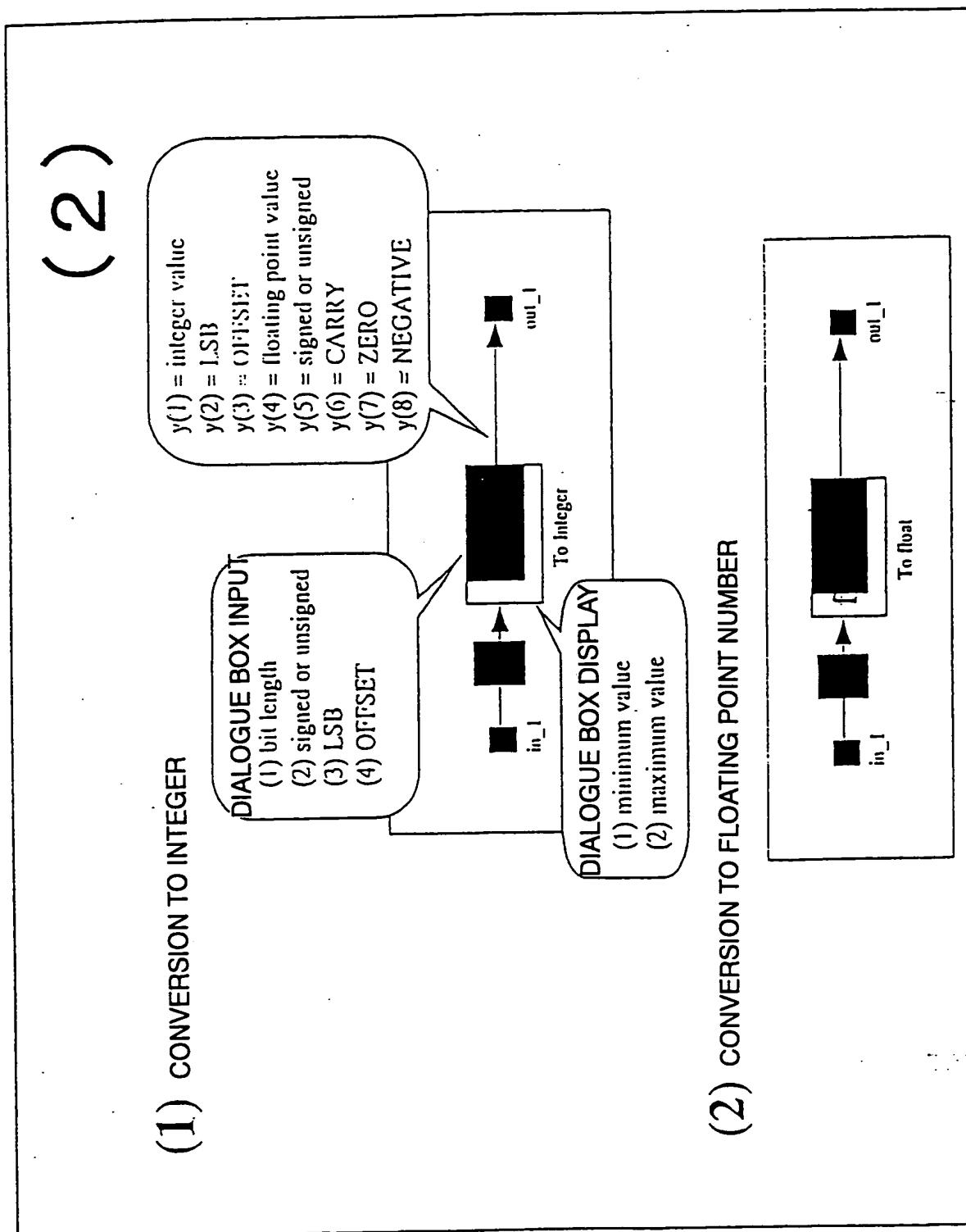


Fig. 12

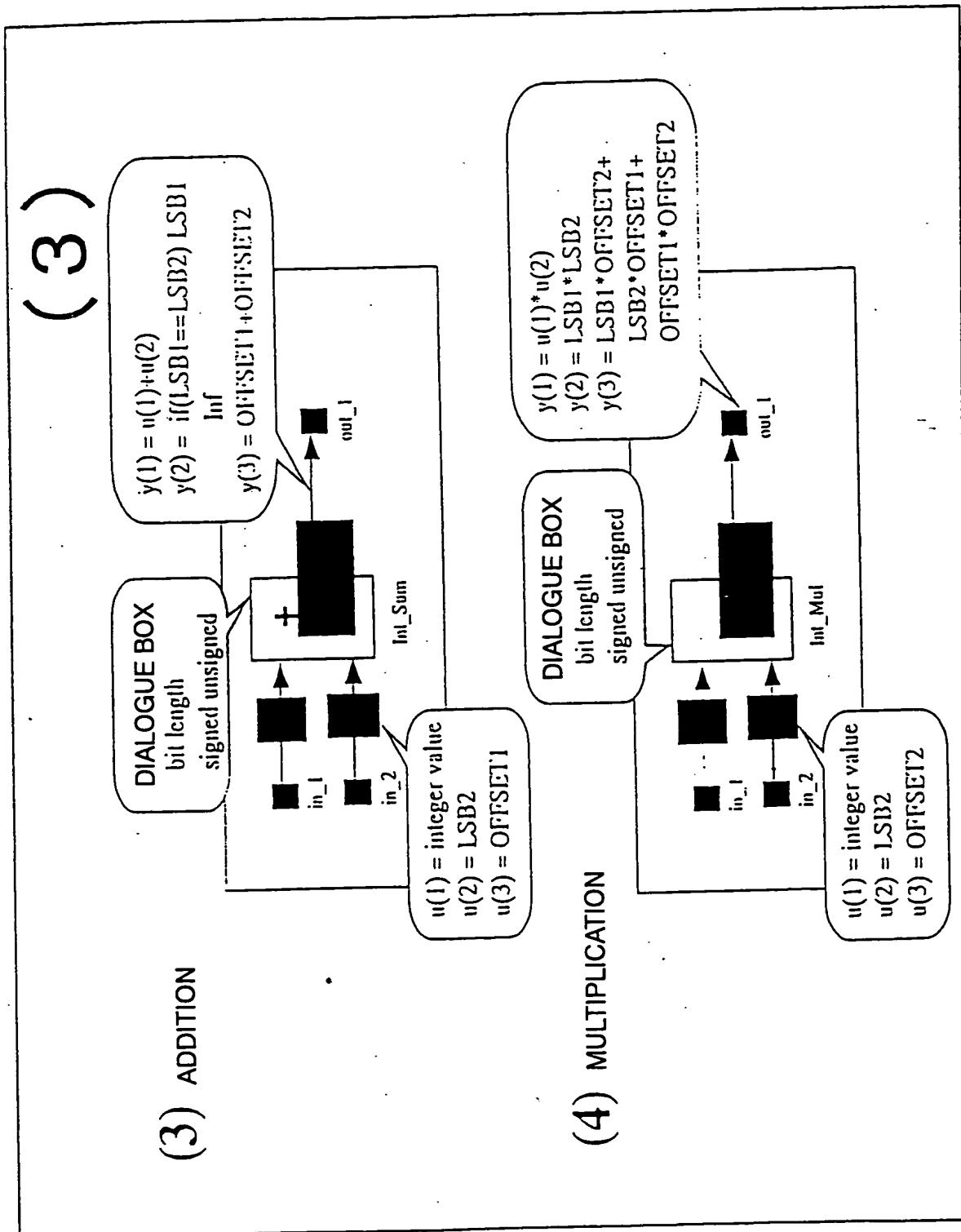


Fig. 13

4

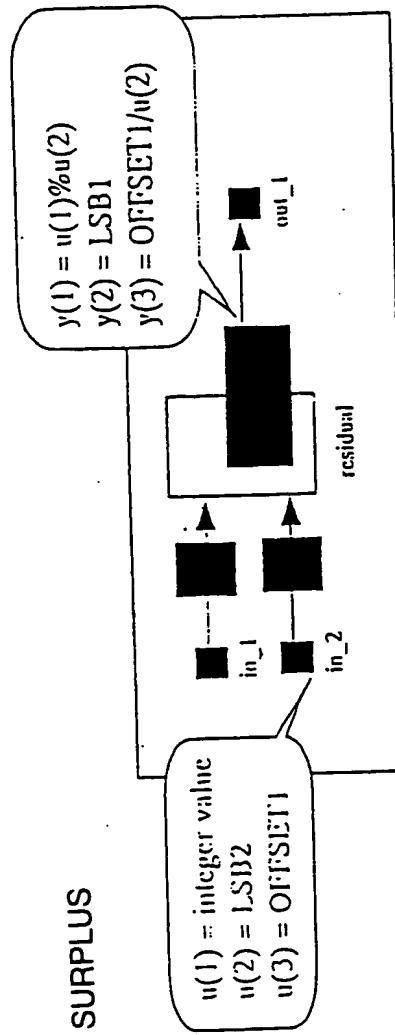
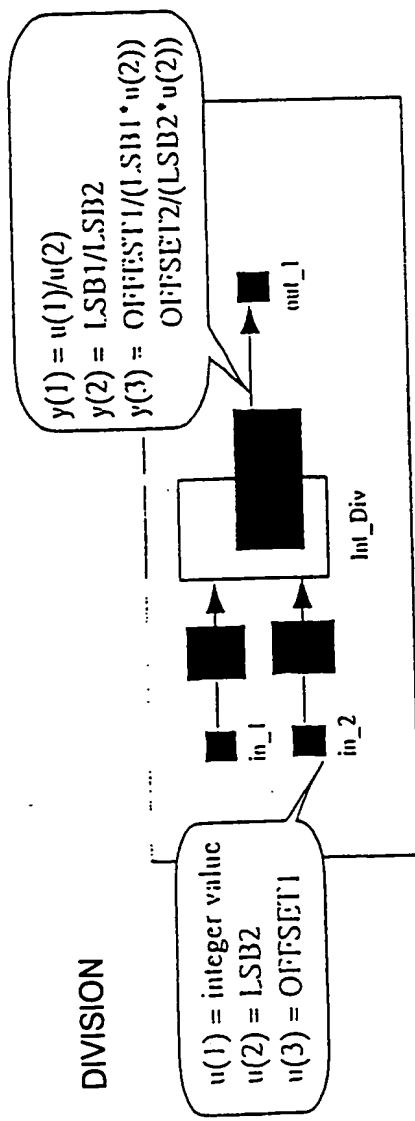
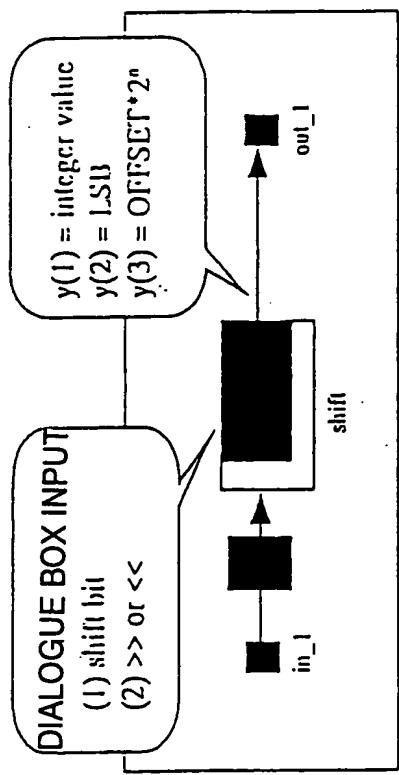


Fig. 14

(5)

(7) SHIFT



(8) (9) TABLE

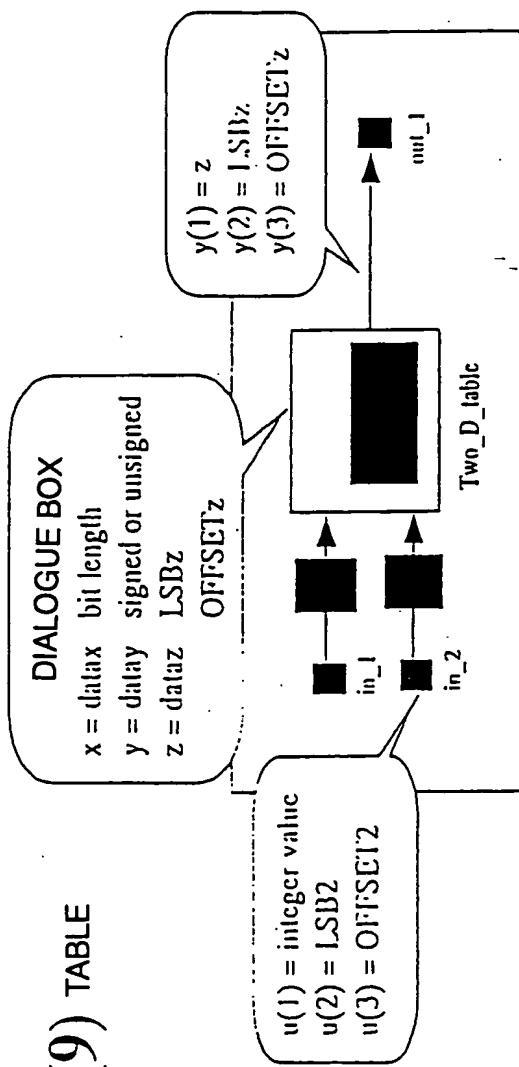


Fig. 15

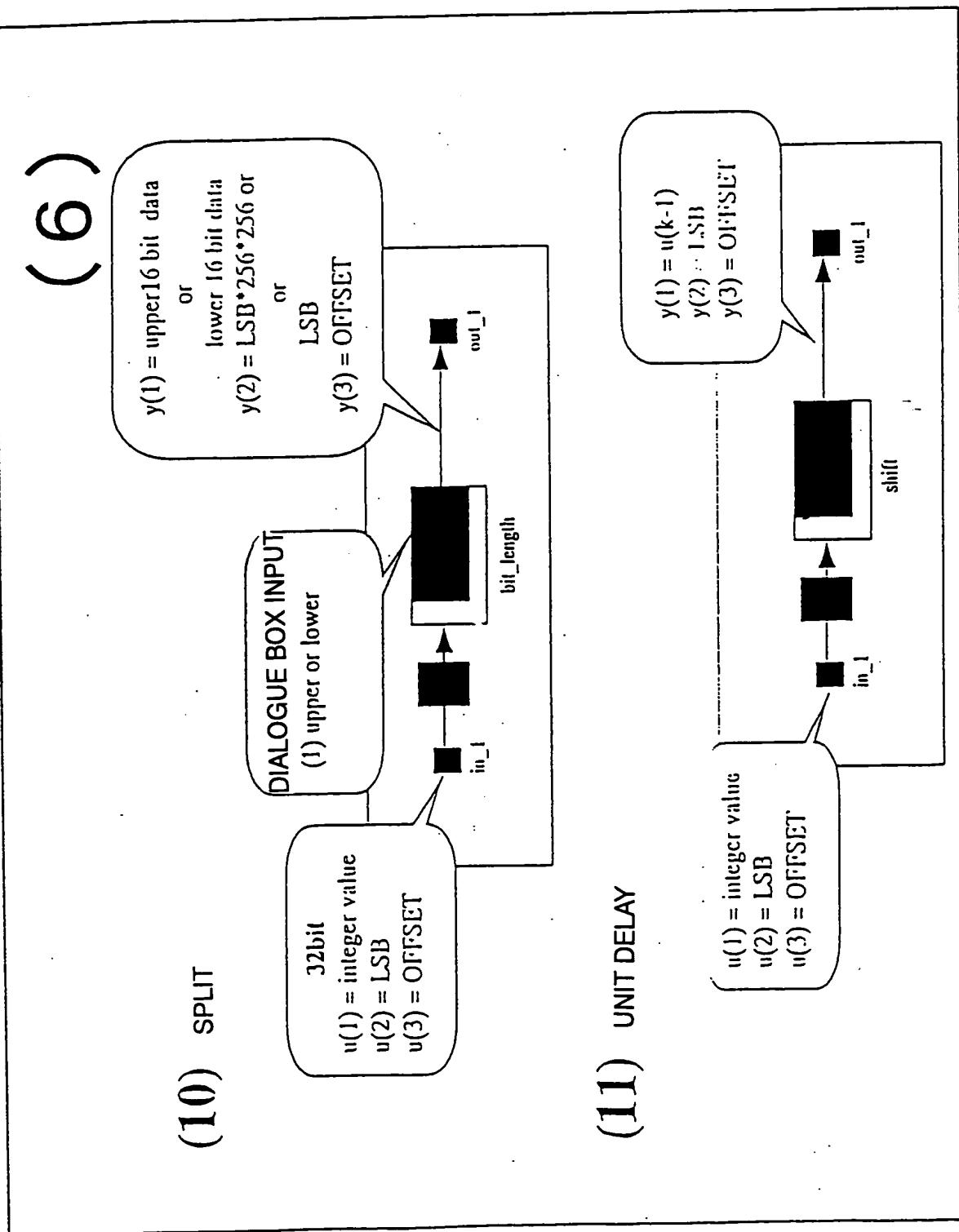


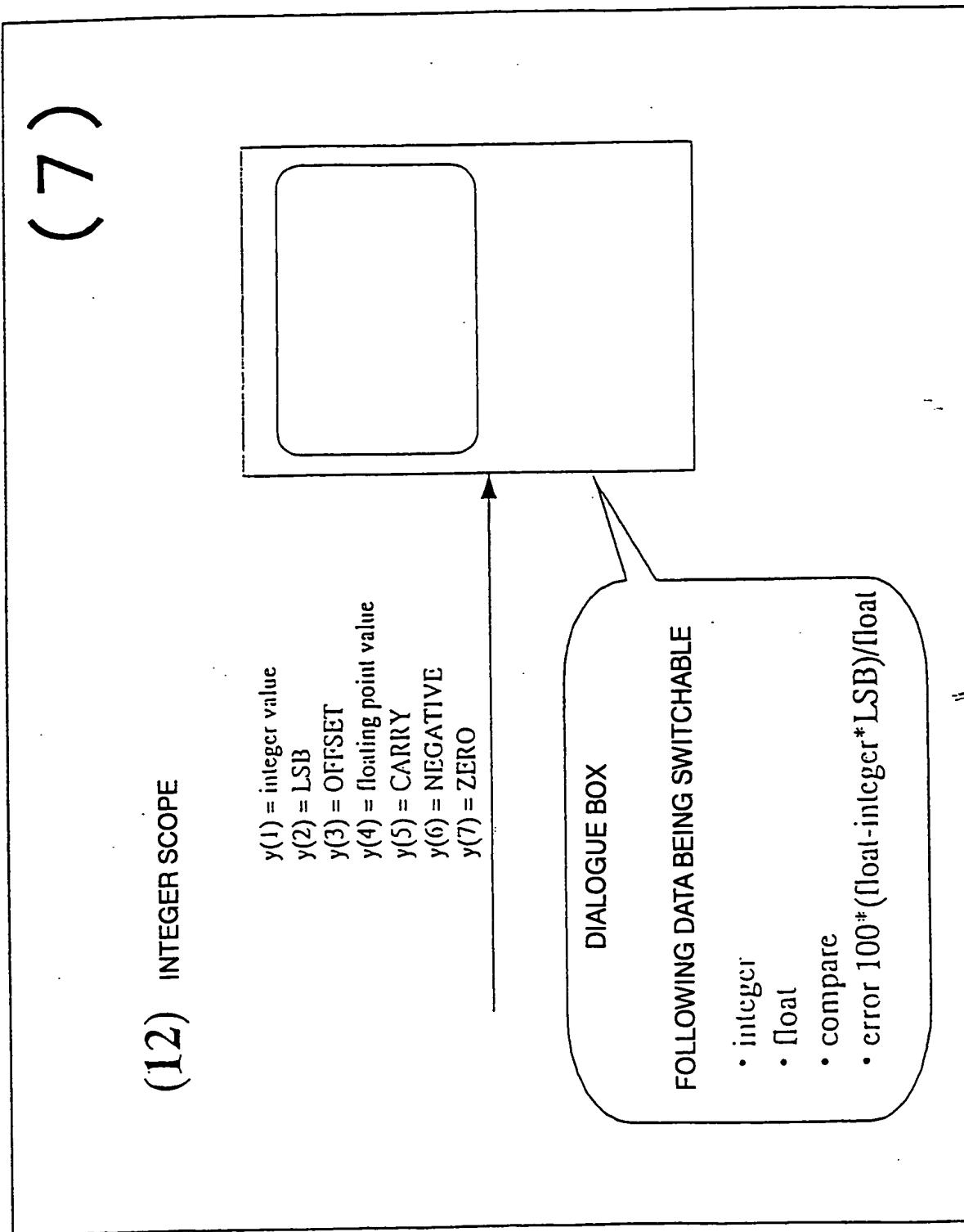
Fig. 16

Fig. 17

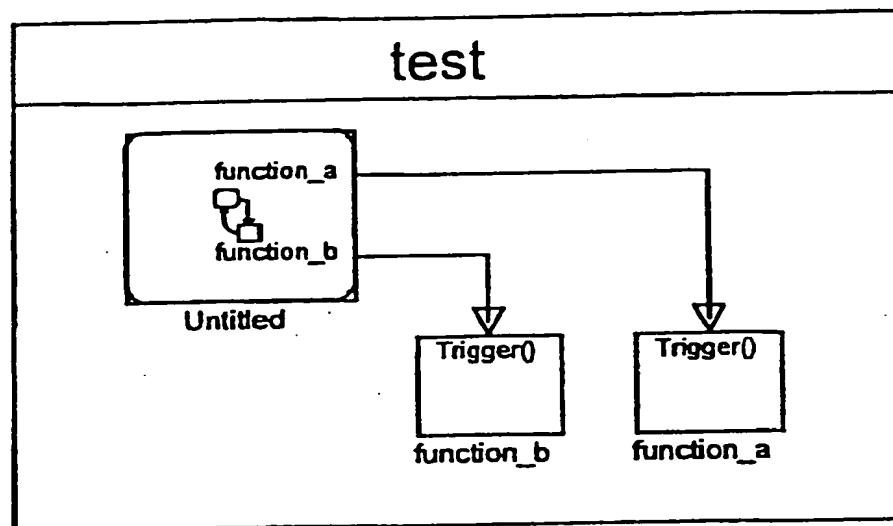


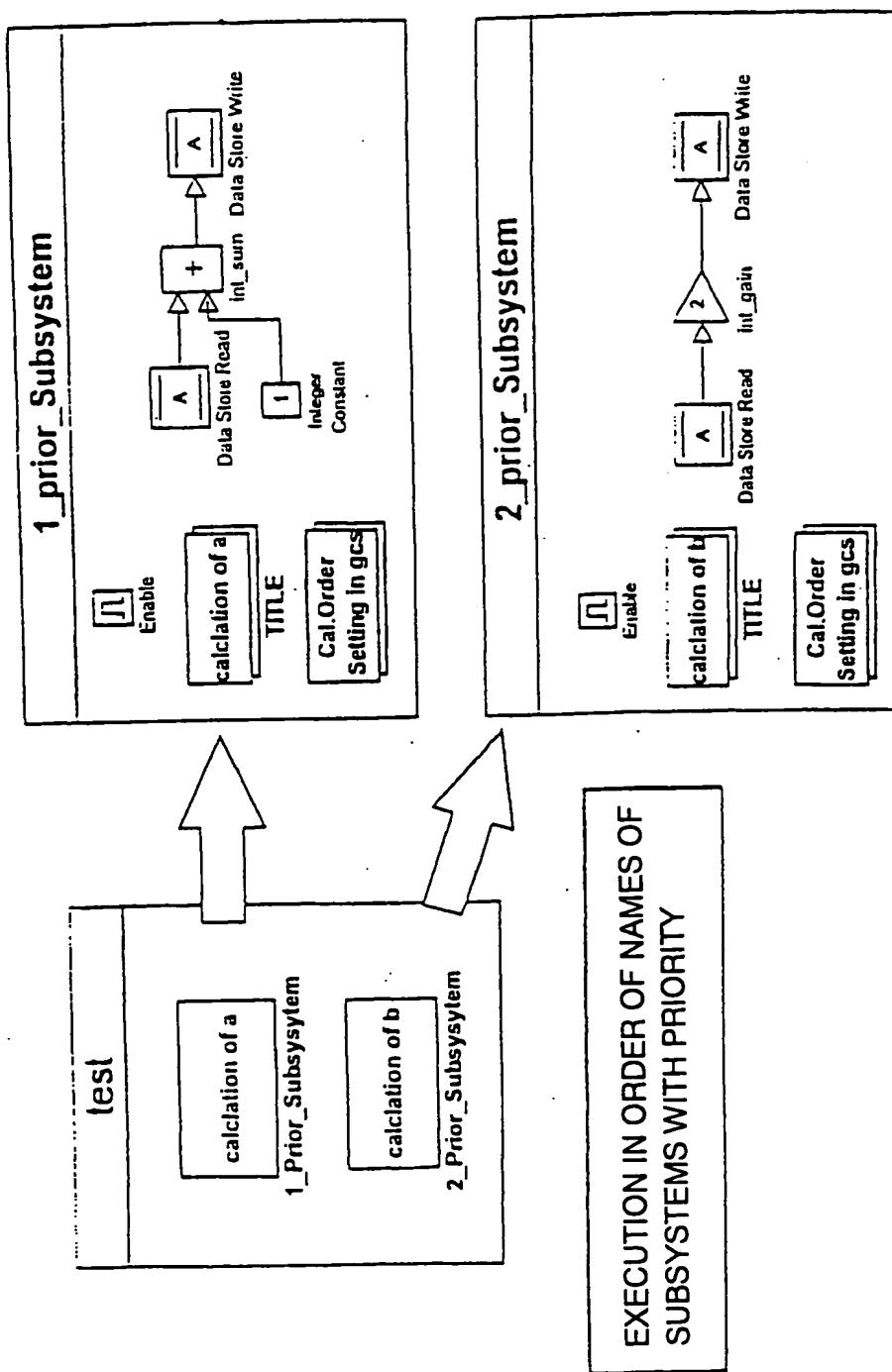
Fig. 18

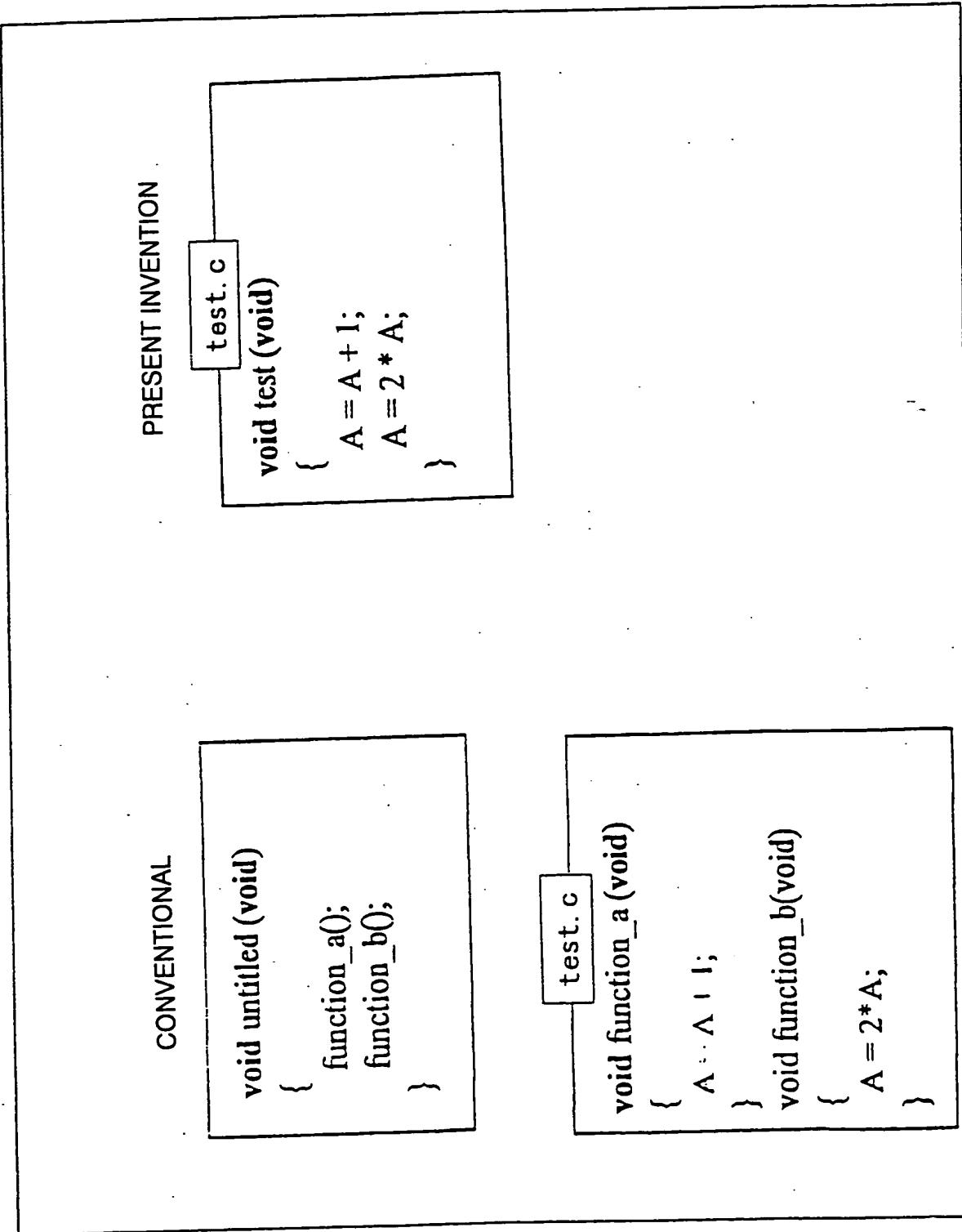
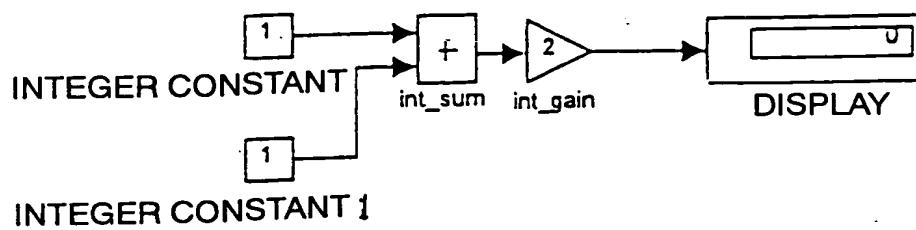
Fig. 19

Fig. 20



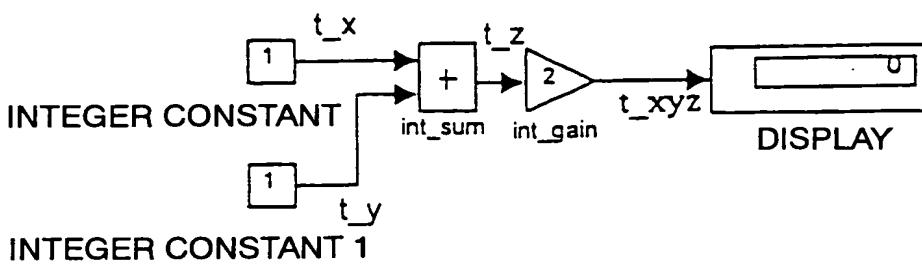
```
void untitled(void)
{
    s16 s1_S_Function;
    s16 s2_S_Function;
    s16 s4_S_Function;
    s16 s3_S_Function;

    /* int_sum : s4_S_Function */
    s4_S_Function = s1_S_Function+s2_S_Function;

    /* int_gain : s3_S_Function */
    s3_S_Function = (s16)(2*s4_S_Function);

    /* (no update to perform in root mode!) */
}
```

Fig. 21



```

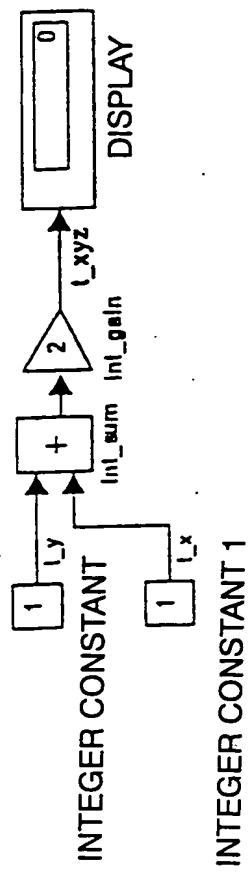
void untitled(void)
{
    s16 t_x;
    s16 t_y;
    s16 t_z;
    s16 t_xyz;

    /* int_sum : s4_S_Function */
    t_z = t_x + t_y;

    /* int_gain : s3_S_Function */
    t_xyz = (s16)(2*t_z);

    /* (no update to perform in root model) */
}
  
```

Fig. 22



WITHOUT GROUPING

```

void untitled(void)
{
    s16 t_x;
    s16 t_y;
    s16 s4_S_Function;
    s16 t_xyz;

    /* int_gain : s3_S_Function */
    t_xy = (s16)(2 * t_x + t_y);

    s4_S_Function = t_x + t_y;
    t_xy = (s16)(2 * s4_S_Function);
}

```

GROUPING

```

void untitled(void)
{
    s16 t_x;
    s16 t_y;
    s16 t_xy;
    s16 t_xyz;

    /* int_gain : s3_S_Function */
    t_xy = (s16)(2 * t_x + t_y);
}

```

Fig. 23

GROUPING

ID	Signal Label
(a) s1_S_Function	t_x

ID	EXPRESSION
(b) s1_S_Function	x1*x2

Fig. 24

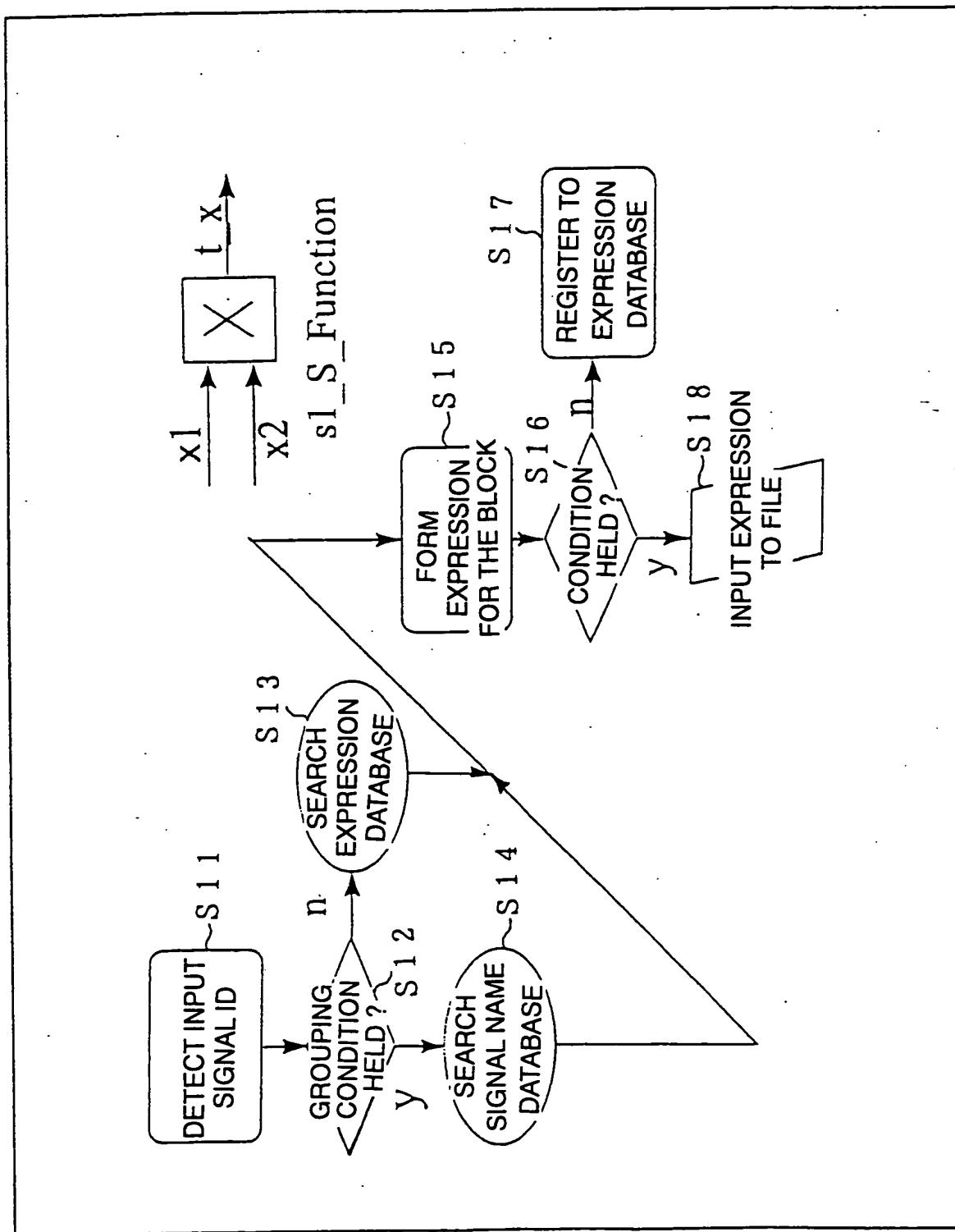


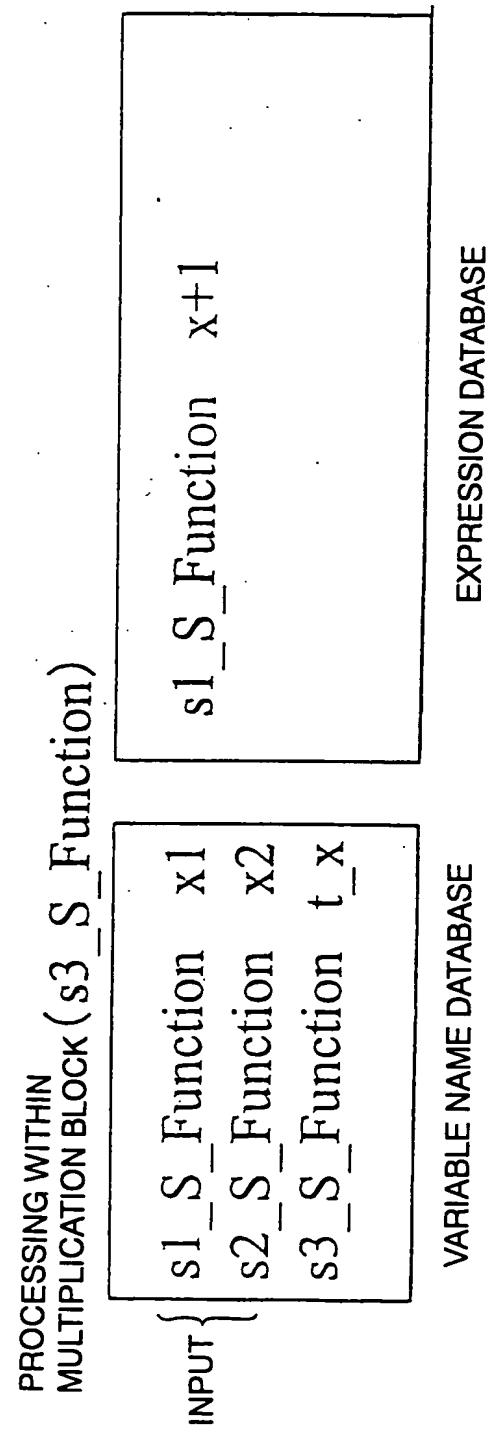
Fig. 25

Fig. 26

